REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	06 NOV 96	Final Report, 1 OCT 95 - 31 MAR 96

4. TITLE AND SUBTITLE
Robust Visual Detection and Recognition of Moving Objects in Real-time

5. FUNDING NUMBERS
AFOSR contract
F49620-95-C-0078

6. AUTHOR(S)
J. Brian Burns, Carlo Tomasi, Stanley Birchfield and Joaquin Salas

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Teleos Research
2465 Latham Street, Suite 101
Mountain View, CA 94040

Stanford University
Computer Science Dept.
Stanford, CA 94305

AFOSR-TR-96 97
0001

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Dr. Abraham Waksman
AFOSR/NM
Directorate of Mathematics and Geosciences
110 Duncan Avenue, Suite B115
Bolling AFB DC 20332-8080

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

This report covers research in three areas of vision technology: the detection, tracking and recognition of objects. The focus of the Phase I research effort was to demonstrate the feasibility of the technology for security and teleconferencing applications. To this end, a system capable of detecting, tracking and identifying human subjects was developed. For object detection and tracking, differential motion segmentation is employed, and both region-based and boundary-based algorithms were developed for this purpose. The recognition module is based on an approach to matching flexible, poorly modeled objects that has the potential speed of traditional indexing methods. The design is an original combination of moment-based image representation, relational match analysis, efficient indexing and robust transformation estimation. The feasibility of all the designs are demonstrated experimentally in the context of the applications.

14. SUBJECT TERMS	15. NUMBER OF PAGES
Visual detection, tracking and identification of objects; security and surveillance; teleconferencing.	41
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
unclassified	unclassified	unclassified	SAR

DISCLAIMER NOTICE

THIS DOCUMENT IS BEST QUALITY AVAILABLE. THE COPY FURNISHED TO DTIC CONTAINED A SIGNIFICANT NUMBER OF PAGES WHICH DO NOT REPRODUCE LEGIBLY.

Contents

1 Introduction

This report covers research in three areas of vision technology: object detection, tracking and recognition. Visual detection is the process of noting and locating something that is different or distinct from its surroundings in some way. For example, objects of interest are often moving relative to the background. Visual tracking is the process of re-locating a relevant object from frame to frame as it, or the camera, moves about. Visual recognition is the process of identifying some unknown portion of the image with something that has been visually acquired and modeled in the past.

These three functions are rapidly becoming feasible engineering goals. When integrated into a single system, their feasibility is enhanced and can be exploited in a variety of practical products that are otherwise difficult to realize.

The first part of this report covers targeted product areas, relevant technical requirements and how vision technology can best satisfy them. The technical focus is in the detection, tracking and recognition of human subjects. Important product areas that can exploit this technology include security and surveillance systems, and teleconferencing.

The rest of the report covers the design and demonstration of relevant vision modules. In each case, the feasibility of realizing the targeted product requirements is demonstrated experimentally.

2 Product areas and objectives

2.1 Relevant product areas

Automatic detection, tracking and recognition of objects can be exploited in many ways. Three areas of particular importance are (1) security and surveillance, (2) teleconferencing and (3) material handling.

With increasingly more complex security operations, larger numbers of cameras and around-the-clock surveillance, an image interpretation process capable of assisting human operators would greatly help the overall effectiveness of security and surveillance products. Users would clearly benefit from automated assistance in the following three areas: (a) alerting a human operator of any distinct, new object in the zones being monitored; (b) tracking an object visually until an operator can deal with it; and (c) aiding in identifying the object using stored visual information. System modules capable of competent visual detection, tracking and recognition are clearly crucial for automated assistance in these respective areas.

In the rapidly growing area of teleconferencing, it is desirable to control the movement and zoom of cameras to optimize the coverage of relevant subjects. In many situations, there may not be an operator available, or there may be more cameras involved than the operator can handle without assistance. The relevance of a visual subject for framing is a function of specific selection by the user and the generic activity of the subject. The latter can be important in situations where the user may want the camera centered on the subject currently talking, writing on a blackboard or gesturing. Clearly, detection strategies based on factors such as visual motion could be useful in these cases, and visual

tracking would generally be useful for continuous control of the cameras. In addition, visual recognition could be crucial in situations where the user desires to pan back to a previously framed subject that is currently out of the frame and may have moved.

Another important application area is material handling. Vision technology could be crucial in situations where objects must be moved and manipulated remotely, and high bandwidth video communication for real-time teleoperation is not possible. In these situations, it would be useful to have automated assistance in visual recognition of the relevant object and tracking of its position as it is moved.

The focus of the current research effort is the development of a visual system for security, surveillance and teleconferencing. To this end, a system is being developed that is capable of detecting, tracking and identifying human subjects.

2.2 Requirements

Given the focus on security and surveillance systems and teleconferencing, the visual processing has the following requirements.

For security and surveillance systems, an automated assistant should be able to detect unexpected objects considered interesting. One powerful, generic criterion for being interesting is having movement independent of camera or scene. The property of independent motion not only contributes to the extraction of an object from the background, but it also is an innately interesting property for objects in security situations. Other features could also be useful for detecting objects from the background, such as color; however, relative object motion is the most crucial property to exploit by a secuity system.

Along with alerting the operator that something has been detected, it is important that the detection process determine the location of the object in the image; this includes extent as well as position. The position and extent are important for determining what data in the image to track in the next frame or attempt to identify. Knowing the extent of the object in the image also helps determine the correct zoom and thus the optimum viewing of the object. In addition, if the identity of the object or the surface it is traveling on is known, the position and extent in the image can be used to estimate the 3D position of the object in the scene. This in turn is important for determining where the object is moving and what other cameras may be able to pick it up.

Once an object has been detected, it should be continually re-located in each new frame. Location determination should again include both position and extent to insure continued tracking and identification.

The ability to detect and track the whole visible extent of the subject is also critical in teleconferencing. It is especially critical in this application, since framing of the subject is the main point of the visual processing.

If a tracked object has been temporarily obscured or has moved nearer to another camera station, it is important that the object be re-acquired when it is again visible from some camera. Visual re-acquisition is a recognition operation, where detected objects in new frames are matched to previously tracked objects. Since the system does not usually have a complete 3D model of the object, it is important that it is capable of identifying an object in the new image using a set of stored images for reference. In addition, since the tracking system does not always have control over the view from which it is

imaging the subject, the set of reference views of an object is often incomplete. Thus, the system should be able to recognize the subject from novel views. Both the security and teleconferencing domains have similar requirements in this regard.

Recognition is also useful as a means of identifying the detected and tracked subject against a more permanent image data-base, such as police photographic files. There is no difference in principal between re-acquisition and identification. There can be, however, a difference in situation that can effect their design. In the former, the reference data will tend to be only seconds old and gathered on-the-fly; while in the latter, the data will tend to be much older and often gathered in a more controlled fashion. For example, in the former, the person's clothing color will still probably be the same; while in the latter, clothing color may be a much less reliable indicator of a person's presence.

Given the targeted applications, the following functions summarize the design requirements for an automated assistant:

- Detect. Detect objects of potential interest, principally utilizing independence of visual motion, and determine their visual position and extent in the image.
- Track. Continue to determine the position and extent of a subject in each new frame.
- Re-acquire. Recognize objects that have been temporarily obscured, or have moved nearer to another camera station.
- Identify. If given pre-stored visual data of objects of interest, match this data to image regions currently being tracked.

2.3 General design

To satisfy the above requirements, a vision system has been designed with three modules: detection, tracking and recognition. It processes data every frame using the following steps:

1. Detection. The position and extent of potentially interesting objects are determined using motion criteria.
2. Tracking. Objects tracked in the previous frame are located in the current frame. If a newly detected object is identified with a tracked one, the position and extent of the new detection is used as the updated position and extent of the tracked object.
3. Re-acquisition. Any object lost in a previous frame and considered still important is searched for in the current frame. The recognition module matches stored images of the lost object to parts of the current frame associated with currently detected and tracked objects.
4. Identification. Objects stored in a permanent database can be searched for in the above manner.

2.4 Objectives for Phase I

Given the above requirements and proposed design, the objective for the Phase I research is to demonstrate to feasibility of the core modules. Specifically, the following is to be demonstrated:

- The detection process is sensitive enough to human motion to reliably detect people in conditions where the camera itself may be moving, as well as other objects.
- The position and extent estimates of the detected object are accurate enough to correctly focus recognition processing and frame the shots by controlling camera pan and tilt.
- Selected objects can be tracked when visible.
- The recognition system can identify an object under change in view. The recognition rate should be good enough under conditions of changing views that when the match results are intergrated across several frames, the recognition rate is very high.
- The processing for each module can be made real-time on conventional machines.

Demonstration of detection and tracking capability is covered in Sections 3 and 4; demonstration of recognition capability is covered in Section 5.

2.5 Integration during Phase II

In the Phase I study, the recognition modules developed and tested in isolation of the performance of the other modules. This is critical for understanding and evaluating the basic behavior of each component. However, detection, tracking and recognition processes can interact in ways that enhance their performance, and this interaction will be exploited in the design of the final prototype in Phase II.

For example, recognition can clearly contribute to a more robust tracking by being used to re-acquire temporarily obscured objects. The current tracking design has been tested without it.

Recognition speed and performance can also benefit from integration with the other modules. By localizing the recognition processing to parts of the image associated with detected, but perhaps unknown, objects, the resources of the recognition system can be more efficiently allocated. Basically, the detected regions of interest could be used to register the pre-stored data of the lost objects, allowing the comparison to be done effectively. Even if the detected regions are not exact, the range of possible registrations could be limited by the detection process. In the current study, the recognition process was tested by simply applying it to the whole image and evaluating its ability to find the target object.

In addition, recognition can clearly benefit from tracking the object being matched. By tracking a detected region from frame-to-frame, the matching of this region to a sought-for object can be performed and integrated over several frames. As the object

moves about, it will often be imaged from different views; this in turn can provide sufficient data for a very confident identification.

Clearly, the integration of the different modules could greatly enhance their individual performances. This will be exploited during the design of the complete prototype in Phase II.

3 Detection and tracking

An actual real-time detection and tracking system based on visual motion analysis has been developed and tested on a 66 MHz Pentium. Its general design and capabilities are discussed and demonstrated in this section. To enhance its usefulness in situations where other moving objects are in contact or are partially occluding the tracked subject, a local motion boundary analysis can be used to properly separate the objects. The design of such an analytic process has been developed, and its usefulness is demonstrated in Section 4.

In situations where the tracked subject is completely lost due to temporary loss in visible contact, it is desirable to re-acquire the subject. This would ensure continuous tracking whenever possible. To this end, visual recognition is required. The design and feasibility study of the recognition module is discussed in Section 5.

3.1 Basic design

As discussed in the previous section, the design objectives for the detection process are that it be: (1) sensitive to visual motion due to human subjects, (2) able to detect the independent subject motion when the camera is also in motion, (3) able to detect separate, multiple moving objects, (4) able to localize the subject in the image and (5) perform at real-time rates.

The current detection process proceeds in three steps. First, using the correlation of the sign of the Laplacian of Gaussian (sLoG) filtered images, the dominant visual motion between the last and current frames is estimated. Then, local sLoG differences are measured between the two images after registering them with the estimated dominant motion. Finally, connected regions of consistently dense difference in sLoG data are extracted. Each region above a critical size is treated as a detected object. Since even small motions of a human subject that are inconsistent with the dominant motion produce significant local sLoG differences, the design is very sensitive to human motion. Figure 1 shows an example of a detected region associated with a person moving relative to the background.

In almost all situations, the camera motion is the dominant motion, thus the motion-based image registration before the difference analysis allows the system to detect small subject motions while ignoring large camera motions. Also, visually separate subjects are extracted as separate regions of sLoG difference. The position and the extent of the regions can be used to localize the subject in the image.

The design objectives for the tracking system is that it can (1) correctly relate detected regions in the current frame to those in the past frame, (2) localize the tracked subject

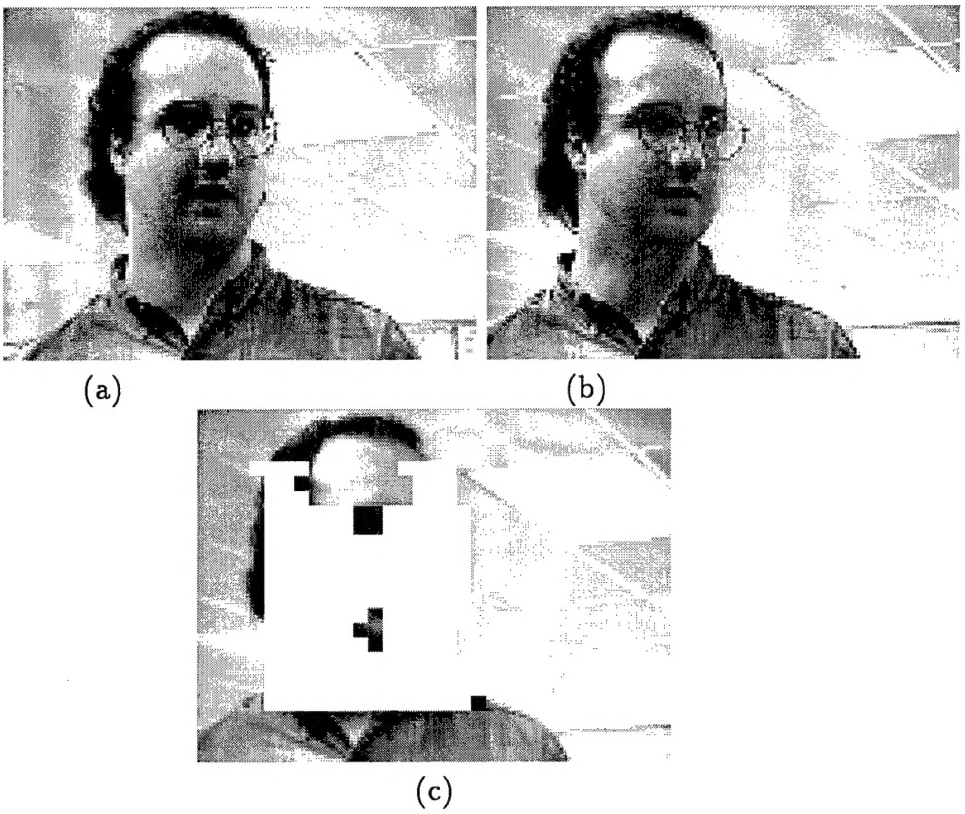


Figure 1: Example of a detected region associated with a person moving relative to the background. (a) First frame, (b) second frame and (c) detected region shown in white. Note background motion due to camera pan.

when it is still visually present but not detected in the current frame, (3) track a subject when there are multiple moving objects, and (4) perform at real-time rates.

The real-time tracking process proceeds in two modes. It first tries to find a detected motion region in the current frame that has significant size and overlap with the tracked region of the last frame. It is possible that no such region exists, even though the tracked subject is still visible. This can occur when the subject did not move relative to the camera. In this case, the tracking process switches to a second mode: the current image is correlated with image data in the tracked region of the last frame, and the point of highest correlation is taken as the new image location. Working in these two modes, the tracking system can localize the tracked object of interest from frame to frame, in spite of camera motion and the presence of multiple moving objects.

3.2 Experiments and discussion

In practice, the detection and tracking module does quite well. Detection and tracking quality were evaluated by using the computed position and extent information to control camera motion. The detection and tracking were considered competent if the camera motion could keep the subject of interest near the center of the image.

The current version has been used to competently track people in real-time for periods of up to a half an hour (54,000 frames), and in the presence of multiple moving objects for up to twenty minutes (36,000 frames). The motions of the tracked subject can be quite large, covering the natural range, and also quite complex, including rapid 3D rotations. Also, the system has been tested for over a dozen subjects and many different backgrounds.

The idea of motion-based detection is similar to the detection strategy used in [21] and the background-subtraction methods in [2]. The contribution of the current work is a design that explicitly compensates for the dominant motion and uses sLoG data, which is insensitive to contrast variation. Dominant motion analysis has also been described in [16]. However, in their work, the motion model is more involved and a practical, real-time implementation using conventional hardware has yet to be established.

Given the above results, the current real-time system appears to go far towards satisfying the application requirements for detection and tracking. Two extensions would greatly enhance the feasibility of these goals. First is the ability to re-acquire the sought-for subject when visibility has been interrupted. This capability is covered by the recognition module, discussed in Section 5. Second is the improvement of motion segmentation by local analysis of motion region boundaries. The current real-time system assumes a single dominant motion that all individual moving objects are distinct from. This can create two problems: (1) the tracked figure may occupy most of the image and thus be providing the dominant motion itself, and (2) multiple figures by be touching or partially occluding each other. In the latter case, the real-time process would tend to mistakenly lump all of the moving objects into one region. By detecting local motion boundaries and using them as a basis for image segmentation, the detection of a dominant motion is not required.

In the following section, designs towards this goal are presented. From experimentation with the real-time tracker and the enhancements demonstrated below, it is reasonable

to conclude that the detection and tracking requirements can be met.

4 Detecting motion boundaries

This section is a report on the results of the investigation of motion boundary detection methods. The goal of this work was to establish whether image analysis of motion boundaries would provide sufficient information for delineating the contours of independently moving bodies in the field of view. While the current research effort has already developed motion tracking technology from live video input, the wide-support spatial integration inherent in the area-based method can generally only be used to determine of the approximate position of moving objects. The study reported in this section proposes to use independent information, that is, motion boundary information, for a more precise delineation of the contours. This is essential if moving objects are to be recognized or classified, and in general for the construction of a detailed map of occlusion phenomena in a sequence of images.

In summary, motion boundary information has proven to be useful. For instance, figure 3 on page 17 compares intensity edges with motion boundaries for a pair of frames in which a person is moving his head sideways. In this example, the camera is stationary, but this information is not used in the boundary detection method. Intensity edges are not direct indicators of motion boundaries, but the latter are often a subset of the former. This implies that motion boundaries, once detected, can be localized with the same, usually high precision with which intensity edges can be localized.

In the course of this research, several different approaches were developed to the solution of this rather difficult but important problem. With each of the methods, both false negatives and false positives occur. False negatives are usually in the form of broken motion boundaries, and false positives are in the form of a "halo" of edges parallel to an actual motion boundary. However, the three methods proposed here work according to rather different principles, and have often complementary properties and performance. It is clear that a combination of these methods can lead to a reliable and flexible motion boundary detection system, which will be developed in the next phase of this project.

In the next section, the problem of motion boundary detection is introduced and motivated, and compared with region tracking methods. Then, section 4.2 introduces three new methods for motion boundary detection. The computational complexity of the methods is analyzed in section 4.3. Section 4.4 discusses the experiments, and section 4.5 concludes with a discussion of the limitations of the approach and directions for further research.

4.1 Motion Boundaries in Image Sequences

Before recognition or reconstruction techniques can be applied to the analysis of motion in image sequences, images must be *segmented* into regions that correspond to bodies moving in different ways. In fact, recognition and reconstruction algorithms usually assume that only the motion of the body of interest is measured, and multiple motions are confusing to most of these high-level techniques. In addition, motion segmentation

yields object boundaries, and these can be used directly for recognition or shape inference.

Many techniques can be used for motion segmentation. This treatment does not consider approaches that require the camera to be stationary, as these involve rather straightforward with image differencing techniques. For more general settings, image motion vectors can be clustered [22] or subdivided into different subspaces [4] that correspond to different 3D motions. Outlier detection techniques can be applied under the assumption that one image motion (usually the background) dominates [16]. Moving bodies are then outliers, and can be segmented out [8]. All these techniques look for *regions* corresponding to different motions, and the motion boundaries themselves are in a sense a byproduct. The resulting motion boundaries are often vague, because the extent of the regions corresponding to different motions can be determined only coarsely. Also, region-based methods must make strong assumptions about the motion within a given region. Rigid motion is one of the most popular assumptions, and all outlier methods require one motion to extend over a large portion of the field of view for acceptable results.

A more direct approach is to look for motion boundaries locally and directly. Evidence for motion boundaries can be found in the discontinuities of a densely sampled image motion field, in the appearance and disappearance of image texture along boundaries that undergo disocclusion or occlusion, and in the deformation of images along occluding boundaries. All these types of evidence were examined, and accordingly three different methods were developed:

- In the *motion discontinuity* method, large variations of image motion across intensity edges are detected.

- In the *lost track* method, a feature tracker is monitored for lost features, which are evidence of occlusion or shearing motion along a boundary. Disocclusions in this method are handled by tracking the sequence backwards in time.

- In the *image strain* method, local image deformation is computed by tracking triangles of point features. Triangles that deform substantially are likely to straddle image discontinuities.

The main advantages of these approaches with respect to region-based methods are a lesser reliance on strong assumptions about motion, and a better localization of boundaries. In fact, only local measurements are used, and only integration of information along motion boundaries is required, rather than in entire regions. The price paid for this greater flexibility and better localization is, not surprisingly, a higher sensitivity to errors in the image motion measurements, since the support of the computation is now reduced from regions to curves. This may partially explain the lesser popularity of these approaches with respect ot region-based methods (but see [19] and [6] for some interesting attempts). Good quality tracking, however, seems to be enough to address this difficulty, as shown in this report.

Since a good and fast region-based method for image motion segmentation has been developed here, it makes sense to integrate this with a boundary-based method for a better localization of motion contours. Although spatial and temporal integration is

crucial, the goal of this brief project was to establish whether the basic image motion measurements are feasible and can be accurate enough for motion boundary detection. Or, stated differently, the goal was to establish how good images are required to be in order to yield good motion boundary estimates.

The following section covers the methods in more detail.

4.2 Motion Measurements and Motion Boundary Detection

In this section, three different methods for the computation of motion boundaries are described. All methods are based on image motion measurements, so the first subsection presents the algorithm for measuring image motion.

4.2.1 Motion Measurements and Their Reliability

Image motion is measured according to the method presented in [17], which in turn is based on [12]. This is a Newton-Raphson style minimization of the sum of squared differences

$$s(\mathbf{d}) = \int\int_{\mathcal{W}} [I(\mathbf{x} - \mathbf{d}) - J(\mathbf{x})]^2 \, d\mathbf{x} \tag{1}$$

with respect to the displacement $\mathbf{d}$. Here, I and J are the two frames and $\mathcal{W}$ is a small window. This method is accurate, and yields displacement measurements to within about one tenth of a pixel of the correct value for typical levels of image noise. However, the method is prone to finding local minima of $s(\mathbf{d})$. Consequently, motion is measured at a coarse resolution first, by running the method on an image that is smoothed and subsampled to, say, half the initial resolution. This doubles the basin of attraction of the correct solution, and smooths away shallow local minima. The displacement from this stage is then used as the starting point for a new computation on the full resolution image.

In order to work correctly, the motion measurement procedure must be applied to regions that are sufficiently *trackable*, that is, that contain sufficient texture. In [17], a method has been defined that determines trackability from a single frame. Specifically, trackable regions are those in which the smaller eigenvalue of the matrix

$$T = \int\int_{\mathcal{W}} \begin{bmatrix} g_x^2 & g_x g_y \\ g_x g_y & g_y^2 \end{bmatrix} d\mathbf{x}$$

is sufficiently large. Here, (g_x, g_y) is the image gradient, and the integrals are computed over the window used for tracking[1]. Intuitively, the two eigenvectors of this matrix represent the two principal directions of texture within the window, and the corresponding eigenvalues measure the corresponding gradient magnitudes. Thus, two large eigenvalues indicate a well-textured, and therefore trackable, feature. One large and one small eigenvalue indicate a strong edge, along which only one component of motion can be measured reliably. Smooth, untextured regions with little intensity variation yield two small eigenvalues: the region is not trackable.

[1]The terms "tracking" and "motion measurement" are used interchangeably in this report.

Details on the selection of trackable windows, as well as of the tracking algorithm, can be found in [17]. In the following subsections, the motion boundary detection methods are introduced.

4.2.2 The Method of Motion Discontinuities

The main guiding heuristic of this method is that motion boundaries often occur along image intensity boundaries, since it is unlikely that foreground and background are exactly of the same brightness. Although their brightnesses may happen to coincide for some amounts of time, over a longer sequence some difference will occur at some point. Of course, this heuristic may fail, in which case some part of a boundary will not be found exactly, but only to the degree of accuracy afforded by the region-based segmentation method that boundary detection is supposed to cooperate with.

Image intensity boundaries can be found by using standard edge detectors [3]. The rest of the computation is then restricted to narrow stripes around intensity edges. Measuring motion exactly along a boundary is hard, since a combination of lens blur, filtering in the camera electronics, and image quantization makes image changes at a motion boundary a complex function of both geometry and photometry that is difficult to predict. Therefore measurements are made slightly removed from the edges themselves, and ignore a stripe of two or three pixels directly on top of intensity edges. If motions on the two sides of an intensity edge are different, the edge is deemed to be a likely motion boundary. A final integration stage can follow intensity edges, collect likelihood information, and attribute a label (motion boundary or not) to the entire edge. Information can also be integrated over time with a Kalman filter for increased reliability. These integration methods are not investigated in this preliminary report.

The approach is different from that in [19] essentially because here measurements are made on the two sides of edges, while they search the entire image for windows with *mixtures* of two motions. Although some interesting results have been presented [1], [18], measuring motion in mixed-motion windows is still rather difficult.

Briefly, to detect motion boundaries in a pair of frames from an image sequence, intensity edges are computed in the first frame. For each edge point, a motion measurement is attempted on each side. If the two measurements are both reliable and sufficiently different, the edge point is labeled an occluding boundary candidate, with a measure of likelihood that increases with the difference between the two image motions.

In the procedure, semicircular windows are used with a radius of 5 pixels for motion measurements (figure 2), which are performed only if the windows on both sides of the edge are trackable.

Given two good motion measurements $\mathbf{d}_1$ and $\mathbf{d}_2$ (two 2D vectors) on the two sides of an intensity edge point, the magnitude of the difference

$$\delta = \|\mathbf{d}_1 - \mathbf{d}_2\|$$

is computed, and represents the likelihood that the intensity edge is also a motion boundary. In a complete version of the method, these values would then be integrated over the intensity edge and accumulated over time with a Kalman filter. In this preliminary

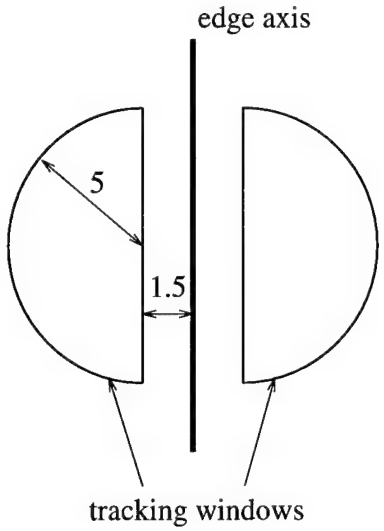


Figure 2: Semicircular supports on the two sides of intensity edges are used for motion measurements in the motion discontinuity method. Distances are in pixels.

study, however, this quantity is simply thresholded for display purposes, in order to see to what extent motion boundary information can be relied upon.

The most critical parameter in this method is the threshold on the motion difference magnitude required to declare a motion boundary. In the experiments, this threshold is set to 1 pixel per frame. In general, the motion difference along a boundary can be arbitrarily small. However, differences below about 0.1 pixels per frame cannot be detected reliably given the performance limits of the tracker. It was decided to stay one order of magnitude above this limit, and to set a threshold at 1 pixel per frame. This implies that slower motion boundaries are not detected *from consecutive frames.* They may still be detected if motion differences are accumulated over time.

4.2.3 The Method of Lost Tracks

In this method, instead of making motion measurements alongside intensity edges, they are made everywhere in the image, and one relies on the fact that along motion boundaries the tracker will fail. Thus, one now relies on measurements that straddle motion boundaries. While [19] proposes a method that looks for bimodal motion distributions, indicating motion boundaries, it is observed that motion at or near a boundary is hard to measure reliably. This low reliability, rather than a nuisance, is exploited here for motion boundary detection.

Failure to track a window along a motion boundary can occur, given enough texture, in two different situations. In the first, parts of the image in the window in the first frame may be occluded in the second frame, so the tracker will either fail to converge, or find a large residue after convergence. In the second situation, the occluded surface translates with respect to the occluding surface in the direction of the boundary. In this case, different parts of the window go to different places of the second frame, and the tracker will fail to find the original image detail in the second frame. In summary, both

occlusions and relative shearing motion along boundaries can be revealed by failure of the tracker to converge to a good minimum of the sum of squared differences in equation (1).

Disocclusions, on the other hand, can be detected based on two different principles. First, a window that straddles the motion boundary will contain background texture in the first frame that moves away from the boundary itself, and is replaced by different image intensities. Second, the tracker can be run backwards in time (from frame 2 to frame 1), so that disocclusions become occlusions, and can be detected in the reverse sequence as described above.

In the method, image features are tracked both forwards and backwards, and points are flagged where the tracker fails in either direction. The following parameters must be set in order to decide whether the tracker has failed or not:

1. Threshold on the residue $s(\mathbf{d})$ after convergence of the tracker.

2. Threshold on the number of iterations by the tracker.

In the experiments, these thresholds are set as follows: if the residue is greater than 5 intensity levels per pixel, or the algorithm iterates more than 20 times without converging to within 0.05 pixels of the previous iteration, the tracker is deemed to have failed, and a motion boundary is declared.

4.2.4 The Method of Image Strain

The two previous methods are in a sense complementary: the method of motion discontinuities relies on motion measurements along the motion boundary, but slightly away from it. In contrast, the method of lost tracks relies on (an indication of poor) measurements for tracking windows on the boundary itself. Both methods, however, require the presence of texture very near the boundary.

The method of image strain attempts to propagate information from farther away when local texture along the boundary is insufficient. In this method, image motion is measured at all points in the image where trackability is sufficiently good. In a typical 480×640 image of good quality, this yields several thousand motion measurements. Tracked points are connected by the edges of a Delaunay triangulation, covering the whole image[2] with adjacent, nonoverlapping triangles. The deformation of every triangle is then measured, defined as the *strain* of its three vertices. Strain, consistently with this notion as used in solid mechanics [5], is defined as follows.

The affine transformation $\mathbf{v}_i' = A\mathbf{v}_i + \mathbf{b}$ is determined that maps (exactly) the three vertices $\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3$ of a triangle in frame 1 into the three vertices $\mathbf{v}_1', \mathbf{v}_2', \mathbf{v}_3'$ of the same triangle in frame 2. Then, the 2×2 transformation matrix A is written by polar decomposition as the product

$$A = RQ$$

of a 2×2 rotation matrix R and a 2×2 symmetric strain matrix Q.

Since rotation and translation are modeled by $\mathbf{b}$ and R, the strain matrix Q describes scaling, possibly different in different directions, and shear. Scaling may be caused by

[2]More precisely, the convex hull of all trackable points.

zooming, and is therefore eliminated by normalizing the perimeters of the initial and final triangles. With this modification, Q now represents shear and changes in the triangle's aspect ratio. Strain is then summarized as a scalar by the quadratic mean of the entries of the symmetric matrix $Q - I$, where I is the identity matrix:

$$\sqrt{((q_{11}-1)^2 + q_{12}^2 + q_{21}^2 + (q_{22}-1)^2)/4}\ .$$

High strain triangles have undergone substantial deformation, and are likely to straddle motion boundaries.

Of course, triangles on steeply foreshortened surfaces are subject to strain as well, and so are triangles on nonrigid surfaces. However, strain along motion boundaries is usually much greater. A threshold is then needed to separate "low" from "high" strain. To select this threshold, a histogram is computed of all the strain values and select the threshold just after the first large peak, corresponding to "low" strain triangles.

4.3 Complexity

If the motion boundary detection methods are to be implemented in real time, an understanding of their computational complexity is required. Feature selection and tracking dominate the computation. For every window, the following operations are necessary.

- Computation of trackability. Assuming that the image gradient is available from previous computations (edge detector), computing trackability requires about $3p$ multiplications and additions, where p is the number of pixels in the window.

- Tracking. Each iteration of the tracker requires $2p$ multiplications and p additions for interpolating image values between pixels, an equal number of operations for interpolating each component of the image gradient, and $5p$ multiplications and additions for the actual displacement computation. On the average, four iterations are required of the tracker. This leads to a total of $56p$ floating point operations to track a p-pixel window. With a 5 pixel radius, the window has 40 pixels, and tracking requires about 2240 operations per window and per frame.

When dense motion measurements are required, special-purpose hardware is mandatory; however, when selected measurements are required, as dictated by the region-based tracker discussed in Section 3, the conventional hardware should be sufficient.

4.4 Experiments

The description of the experiments starts with figure 3, which shows two frames of a sequence in which the person in the foreground (Joaquin Salas) moves his head laterally.

4.4.1 The Motion Discontinuity Method

In order to apply the motion discontinuity method, intensity edges must be found first. Figure 3 (c) shows the output from Canny's edge detector for the first frame. Edges in

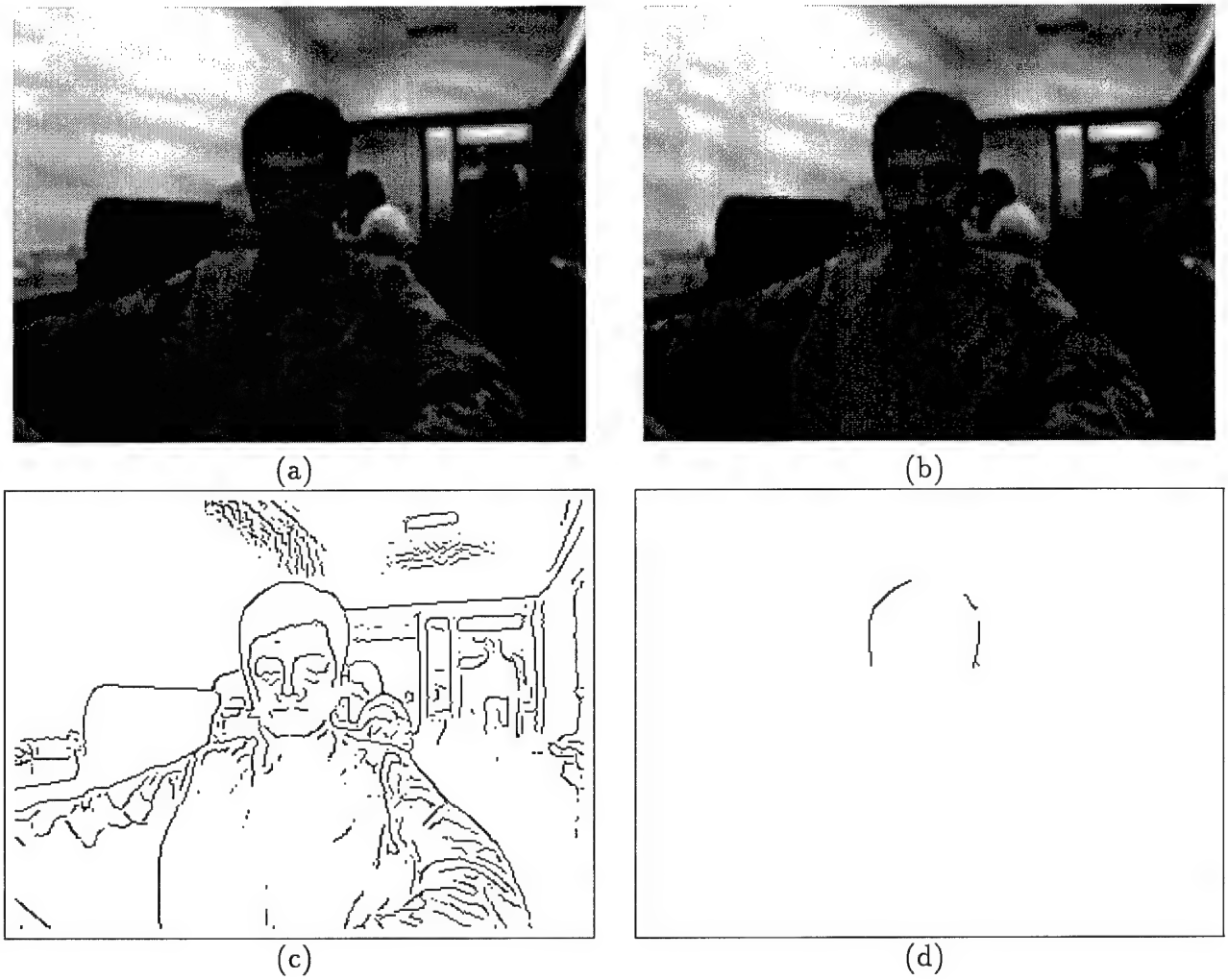


Figure 3: (a), (b) Two consecutive frames of a low-resolution (320×240 pixels, inaccurate focusing) image sequence. (c) Canny edges from the left image. (d) Motion boundaries detected by the motion discontinuity method.

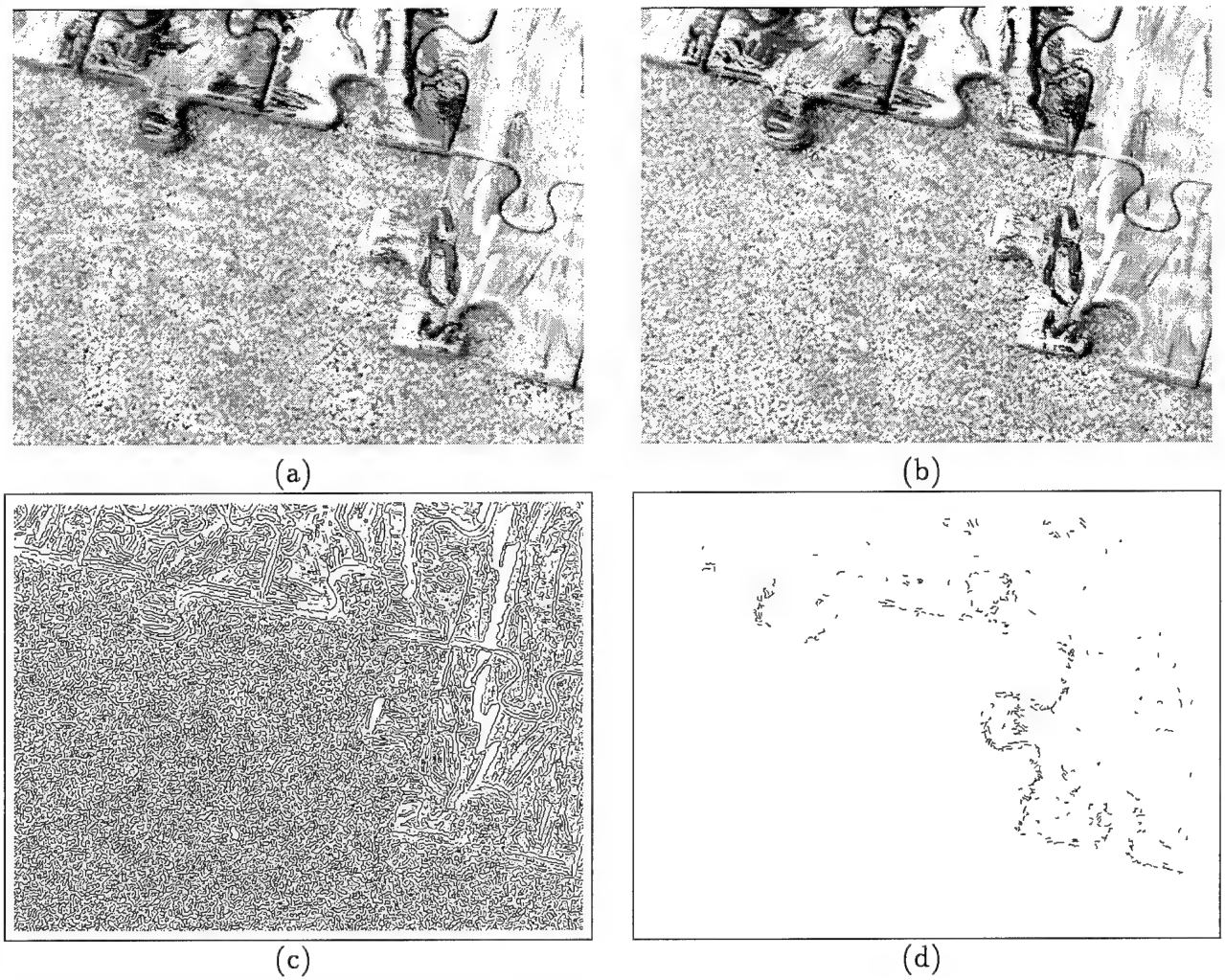


Figure 4: (a), (b) Two consecutive frames of a good quality, 640×480 pixel image sequence. Focusing is good, and frames are not interlaced. (c) Canny edges from the left image. (d) Motion boundaries detected by the motion discontinuity method.

the second frame are about the same. Notice that edges are found around the entire head of the person, although broken at points.

Figure 3 (d) shows the motion boundaries detected by the motion discontinuity method. Insufficient texture was found at the top of the head, both on the hair and on the back wall, so only part of the motion boundary is found. People moving in the background generate no motion boundaries, since their motions are very small (below one pixel per frame).

At first sight, the frames in figure 4 would seem to be an easier case. In fact, texture along the boundaries is sharp and fine, and the tracker ought to work well. And indeed it does: 9300 points are successfully tracked, and the accuracy of the measurements is good.

However, although there are very many edges (see figure 4 (c)), the important ones are not there: in other words, the boundary between the jigsaw puzzle and the table

are not well delineated. At some points, the edge is missing altogether, at others it is confused by the nearby texture and winds around the granular elements on the table surface. Accordingly, motion measurements are made along edge elements that do not correspond precisely to the boundary, with rather erratic results, shown in figure 4 (d).

Another problem evident in figure 4 (d), in addition to broken and erratic motion boundaries, is the presence of several false positives on the surface of the jigsaw puzzle. These are caused by highlights reflecting on the curved parts of the glossy surface of the puzzle. However, these are not, strictly speaking, errors. In fact, the highlights do move with respect to the surface on which they lie, so their contours are motion boundaries. It is important to point out that highlights are often easy to detect, because they correspond to high intensity values, and often to saturated pixel values.

The conclusion from this experiment is that motion boundaries cannot always be expected to lie along intensity contours. Or, at least, they are not along the contours found by Canny's edge detector. In principle one could use a texture edge detector instead, but no sufficiently accurate detector of this type seems to exist. Better intensity edge detectors (e.g., [23]) may do better, but they may still not do well enough for the purposes of this research.

Figure 5 shows a few frames from a low-resolution, interlaced sequence apparently taken without electronic shutter. Next to each frame, the results of the motion discontinuity detector are shown. The camera used for this sequence produces interlaced video, which combined with the lack of an electronic shutter yields two rather different fields in each frame. Figure 6 is a magnified detail of the first frame, which shows the interlace effect. This effect makes good motion measurements difficult.

In the first frame shown in figure 5 (frame 5 of the sequence), only the head of the person in the foreground is moving, and boundaries are detected reasonably well, although a few false negatives and positives can be seen. The second frame (frame 12) shows a person walking into the field of view. This event happens to coincide with the moment in which the head in the foreground has finished one oscillation and starts to move in the opposite direction, so no head motion is detected at this instant.

A couple of frames later (frame 14, third frame shown in figure 5), not much has changed. Notice, however, that the boundaries of the walking person, although correctly detected, are accompanied by a "halo" of edges from the books in the background. This effect is related to the rather wide support used to compute motion along intensity edges. In fact, books in the background produce strong edges, which are therefore candidates for motion discontinuity detection. When motion is measured along these edges, the tracker windows extend to the walking person, thereby "capturing" some of that motion, and making the book edge appear as a motion boundary. This effect is already present to some degree in frame 12 (second frame in figure 5), but is even more evident in frame 14 (third frame in figure 5), where two people walking in opposite directions compound the effect. Poor texture along the motion boundaries made the algorithm miss the second walking person altogether.

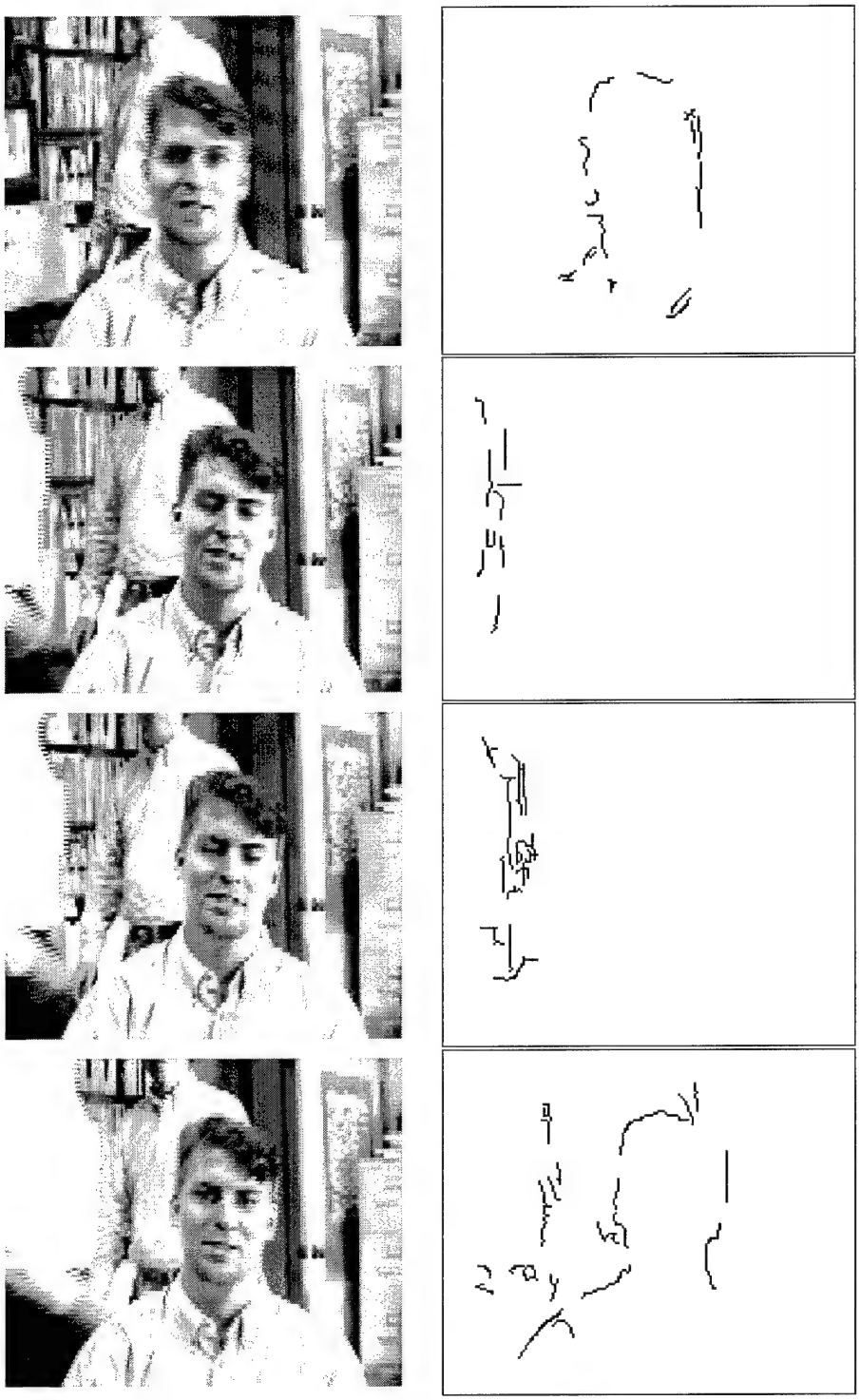


Figure 5: Four salient frames of a low-resolution, 320×240 pixel, interlaced image sequence taken without electronic shutter. Focusing and lighting are poor.

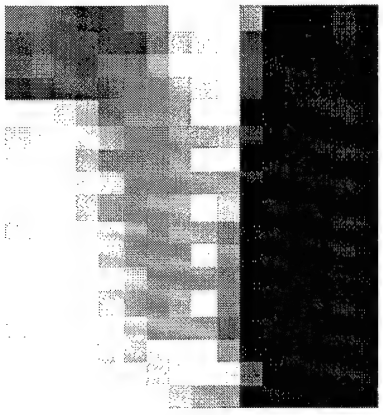


Figure 6: A detail from the first frame in figure 5, showing the effects of field interlace. Even scanlines are displaced by as much as three pixels with respect to odd scanlines, because they are taken 1/60th of a second apart.

4.4.2 The Lost Track Method

The jigsaw example in the previous subsection (figure 4) exposed an important shortcoming of the motion discontinuity method. In fact, this example shows that motion boundaries do not always correspond to intensity edges detectable with standard edge detectors. The lost track method, on the other hand, does not require intensity boundaries, as it measures motion wherever in the image there is sufficient texture for trackability. Figure 7 shows the result of applying this method to the two frames in figure 4. Essentially no false positives were found, and most of the boundary was correctly recovered.

False negatives occur in the upper left part of the image, where relative motion is nearly parallel to the motion discontinuity. In this area, texture on the jigsaw pieces is insufficient. This fact emphasizes an important difference, regarding the lost track method, between motion boundary elements in which relative motion is parallel to the boundary itself (the *parallel motion* case), and elements in which it is not (the *nonzero normal motion* case). In the latter, parts of the background actually disappear from the image (when tracking either forwards or backwards in time), while in the former they do not. Consequently, the lost track method works well in the nonzero normal motion case if at least the background is textured. In fact, relative motion hides background texture, and this then leads to poor tracking performance, as expected. In the parallel motion case, on the other hand, both the foreground and the background must be textured. In fact, when only one side is textured, the tracker will follow the other side without penalty, and no tracking failure occurs to reveal the occluding boundary.

4.4.3 The Image Strain Method

Both methods discussed so far perform poorly if texture is not present very close to the motion boundary. The image strain method, on the other hand, will yield boundary information even if texture appears far from the boundary. In fact, every part of an occluding boundary must belong to some triangle of the Delaunay triangulation. That

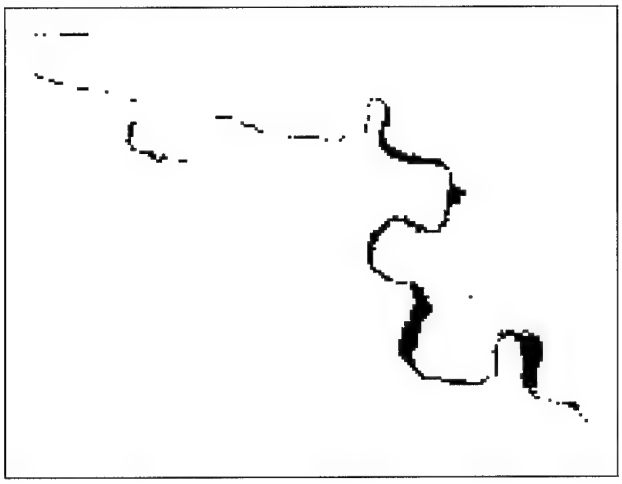


Figure 7: Motion boundaries detected in the frame pair in figure 4 by the lost track method.

triangle has its vertices on surfaces that move differently, and is therefore subject to strain, even if the vertices are far from the motion boundary. Consequently, the occluding boundary is revealed. Of course, where texture is distant from the boundary, the triangles are large, and the positional uncertainty of the boundary is thereby increased: It is only known that in a given triangle there is a boundary, but it is not known where in the triangle it is. Edge detection within the triangle may reveal an intensity discontinuity, to which the motion boundary can be associated. This possibility has not been explored yet.

Figure 8 (a) shows the Delaunay triangulation of the 9300 trackable points found by the feature selection algorithm. Triangles are very small on the fine texture of the table, and they are somewhat larger and less regular on the jigsaw puzzle. In figure 8 (b), the triangles of figure 8 (a) are painted with gray values proportional to their strain values (darker triangles represent higher strain values). Notice that the boundary is all covered by high-strain triangles. However, strain appears also on the jigsaw puzzle, and is caused by the motion of highlights.

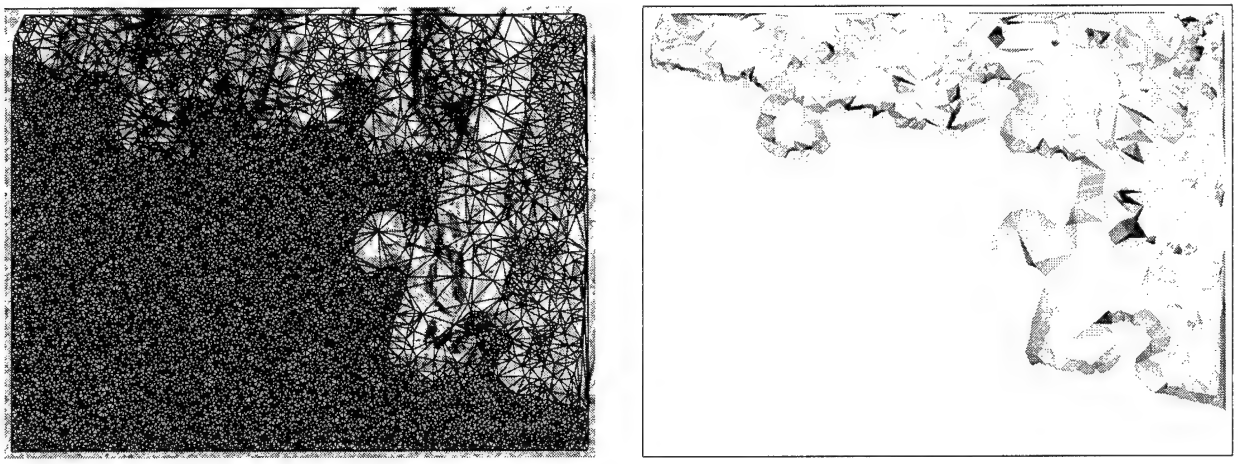
Because of the complete connectedness of the sequence of high-strain triangles that in figure 8 (b) cover the correct motion boundary, this method seems to have great promise. Ways to determine when strain is caused by highlights in reference to figure 4 (d) are discussed in a previous section.

As an additional example, figure 9 (c) shows strain for the frame pair in figure 9. For comparison, figure 9 (d) shows the motion boundaries computed from the same frame pair by the motion discontinuity method.

4.5 Motion boundary detection conclusions

In this project, the feasibility is examined of motion boundary detection. The goal was to answer the question, whether measurements could be made in an image sequence that would directly reveal motion boundaries.

Overall, the answer is positive. Although both false negatives and false positives



(a) (b)

Figure 8: (a) Delaunay triangulation of the 9300 trackable points ound in the first frame shown in figure 4. (b) Image strain.

occur with all the methods, a large fraction of the motion boundaries are correctly detected. Moreover, methods have been pointed out for addressing the various problems encountered. In particular, a combination of at least two of the three methods seems to be promising, because of their complementary features. Also, integration over intensity contours and stochastic estimation over time can be used in order to exploit more global consistency information than raw local measurements can convey.

The jigsaw puzzle images of figure 4, together with motion boundaries computed by the motion discontinuity method (figure 4 (d)) and the lost track method (figure 7), as well as the high-strain triangles of figure 8 (b) shows strengths and weaknesses of the various approaches. It is believed that the strain method shows particular promise because it yields adjacent triangles, not simply points. A motion boundary is then a long, connected sequence of such triangles. Determining the motion boundary, or rather the triangles that cover it, can now be cast as a global optimization problem. This possibility will be invested in future research. In addition, the contours of the dark region found by the lost track method (figure 7) are detailed traces of the motion boundaries in the two frames. While the strain method is better at finding a connected boundary region, the lost track method can be employed subsequently in order to localize the boundary with higher accuracy.

5 Recognition

This section presents the design and feasibility studies for the recognition module. This module is to be used to re-acquire an object of interest that the system has lost track of, possibly due to being temporarily obscured. Recognition can also be used to identify the tracked subject from a permanent data base. Examples of such data bases could be police photographic files for security applications, or private photo files for teleconferencing.

Practical recognition is feasible if the recognition success rate is satisfactory, given

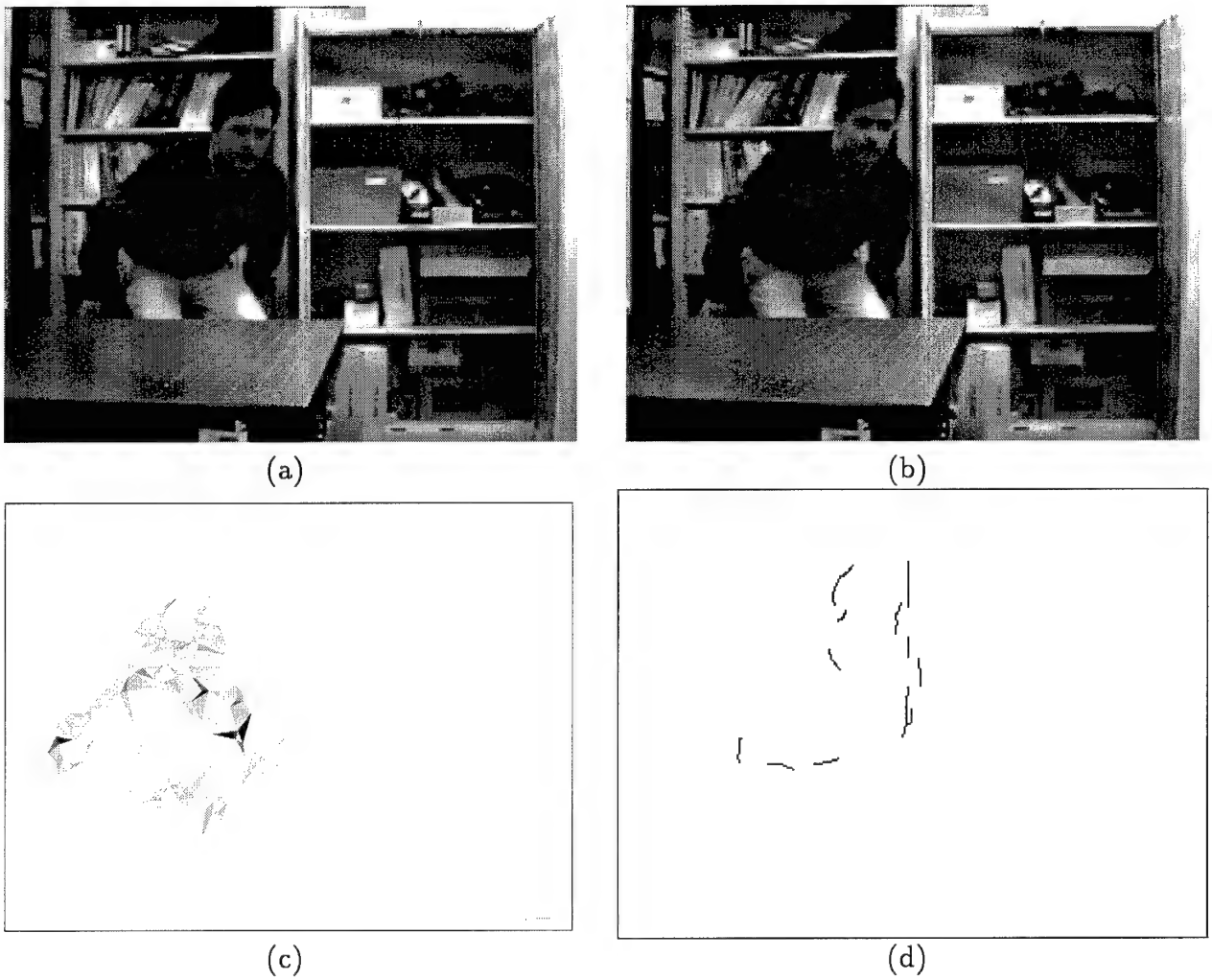


Figure 9: (a), (b) Two consecutive frames of a medium quality (640×480, accurate focusing, poor lighting) image sequence. (c) Image strain. (d) Motion boundaries detected by the motion discontinuity method.

realistic input, and the system can be made fast enough. The feasibility of the design is demonstrated at the end of this section; it is based on matching experiments involving over 300 images.

5.1 Requirements and key properties

Different application domains stress different aspects of the general recognition problem. The recognition requirements are key for security and teleconferencing applications:

1. The system must be able to utilize object information in the form of raw image data.

2. The system must be able to utilize a sampling of views of the sought-for object that may be incomplete.

3. The system must be able to match the object data to potentially disrupted input image data.

4. The system must be fast enough to identify subjects potentially moving out of the range of observation.

These points are expanded in the following discussion.

(1) Object information is in the form of raw image data. There is generally no precise 3D model of the object being sought-for; human appearances vary a lot due to clothing, shape and articulation. Also, it is difficult to find simple physical features of humans whose projection in the image can be precisely predicted and detected. Such 3D models and physical features are often exploited in the design of recognition systems for industrial automation. For example, special features such as straight edges or elliptical arcs are often readily predicted given 3D models and detected in the image. In the targeted applications, there are no such exact models or features. Instead, object reference data is generally derived directly from raw image samples of the sought-for subject and the features used to recognize the object should be simple, generic functions of the 2D image data.

(2) The sampling of views of the sought-for object may be incomplete. For example, police photo files may only have full-face and profile views of a person, and the security system may only be able to see a suspect from some intermediate view. Another important example for both applications is re-acquisition. In many situations, re-acquisition must be done by matching models of the object that are made on-the-fly, using whatever view samples the subject has presented to the camera. Given possibly sparse view samples, it is important for the recognition process to apply a flexible model of the object's appearance. Instead of attempting to match rigid 2D templates of the object's image, the matching process must be able to tolerate unmodeled global distortions and transformations.

(3) Image data disruptions must be handled, such as partial occlusion of the object, lighting change and moving facial features (e.g., talking, blinking). To handle such disruptions, it is important that the recognition module utilize many features that are spatially distributed about the object's image. In this way, parts of the object that are

temporarily obscured, or too distorted (e.g., a yawning face), do not effect the overall recognition. The use of many features, possibly hundreds, can help increase the flexibility of the matching process.

(4) The recognition module must be fast. The detection and tracking of independently moving image regions is performed at frame rate by the module discussed in Section 3. With reliable frame-to-frame tracking, the identification of these regions does not need to occur at frame rate. It must occur fast enough to alert operators and allow the camera to frame the most important object, when multiple regions are detected. Given typical human motion and camera distances, any tracking and operator decisions require the recognition to occur within one or two seconds. Another factor in the time requirements is that the analysis is generally only required in the tracked regions; this can speed up the processing by several times. However, since recognition performance is often a function of the number of images used to identify the object, the faster the recognition process, the more frames can be tested and the greater the confidence in the match. Thus ideally, the recognition system should be able match an object model to a tracked region several times a second.

In summary, this analysis of the problem implies the following key properties for the recognition module:

1. Image-based. Features used for recognition should be simple, generic functions of the raw image data.

2. Flexible. The features and matching process should be able to tolerate unmodeled global transformations of the image. For example, the recognition module should be able to identify a head directly facing the camera with the same head seen in a three-quarter-face view.

3. Spatially distributed features. To ensure a confident match, the system should not depend on any single subset of the features.

4. Fast. The recognition system should be able match an object model to tracked region several times a second.

5.2 Design given requirements

This section presents a recognition module design based on the above properties. The basic design follows three steps: (1) measure image features, (2) rapidly detect rough matches between image and modeled object, and (3) refine and evaluate match via object localization.

5.2.1 Features

Given the above discussion, the features used to recognize objects should be (1) simple, generic functions of the image data, (2) reasonably stable with respect to global transformations such as 3D rotation of the object, (3) capable of representing independent pieces of information at many different locations distributed about the image of the object, and (4) fast to compute.

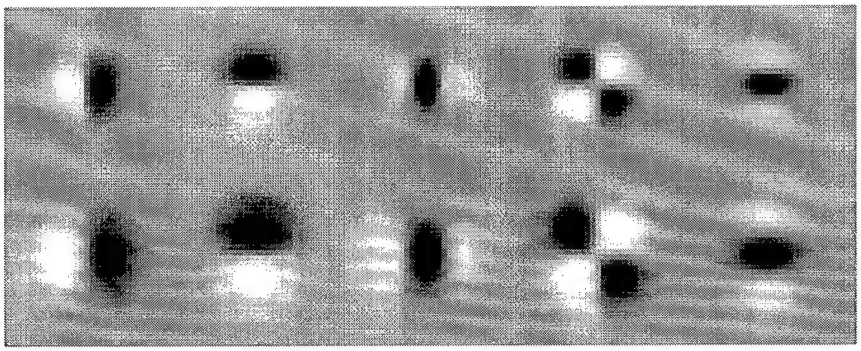


Figure 10: Examples of Gaussian-weighted moments (derivatives of 2D Gaussian) at two scales. Bottom row is one half-octave larger than top, and the derivatives are (left-to-right) g_x, g_y, g_{xx}, g_{xy} and g_{yy}.

These properties can be satisfied by using Gaussian-weighted image moments, especially when they are suitably normalized, sampled at Laplacian of Gaussian peaks and augmented by semi-local geometric features.

Image moments are an infinite series of terms that describe the distribution of some type of value in the image [15]. The example explored in this work is image intensity, though other types of image data such as color or texture can be similarly represented. The different terms in the series characterize different aspects of the intensity pattern in the image, and they tend to be very efficient in their representation: the first few terms characterize much of the overall pattern. They are simple and direct functions of the data, meeting the first requirement. Also, since the lower terms represent coarse features of the pattern, they are relatively stable with respect to transformations of the images. In fact, they can be explicitly normalized with respect to changes in brightness, contrast, rotation in the image and other transformations. Thus, using the first few image moments, normalized, helps to satisfy the second requirement.

Gaussian-weighted moments have the additional advantages of locality and efficient implementation. By modifying the mean of the Gaussian weighting function, the position of the data contributing to the moments shifts in the image, and by modifying the Gaussian scale factor (σ), the size of the contributing region also changes. Thus the position and size of the image locality being measured can be varied. Figure 10 shows all of the first and second Gaussian-weighted moments for two different scales. The bottom row shows moments at a scale that is larger than the top row by $\sqrt{2}$. What are being shown in the figure are the convolution kernals associated with the moments: the light regions are positive coefficients of the kernals, and the dark regions are negative. The distribution of the coefficients define the position, scale and aspects of the intensity pattern that the moment is tuned to. By using multiple moments centered at the same position, the intensity pattern of that locale can be represented.

By measuring the moments over a range of positions and scales, the third requirement is satisfied: a large set of spatially distributed features are available for recognition. Along with being localized, Gaussian-weighted moments help satisfy the final requirement: they can be efficiently implemented. They are simple, linear combinations of the derivatives of the Gaussian filter, which in turn is separable and be rapidly computed. In fact,

the real-time detection and tracking module discussed in the previous section already computes two of the second derivatives everywhere in the image in a few milliseconds on a 66 MHz Pentium.

The moments computed at two close positions and scales are highly correlated. To compute a minimal set of measurements that are reasonably independent, the moments should be applied at selected positions and scales. The peaks of the Laplacian of the Gaussian (LoG) of the image are reasonably stable as the image changes, and if the scale of the peak detection is matched to the scales used to compute the moments, the spatial sampling induced by the peak locations is a good compromise between maximizing the number of feature measurements and their independence. For the scales used in the current study, restricting the moment measurements to the peaks reduces the number of moment measurements from tens of thousands to a few hundred. This increases efficiency dramatically and still provides hundreds of features for recognition.

Thus Gaussian-weighted moments measured at LoG peaks helps satisfy all of the above feature requirements. The usefulness of moments can be further enhanced by semi-local analysis. This is a combination of localized moment measurements and a geometric analysis of multiple measurement points (peaks) over a neighborhood of the image. The geometric arrangement of the neighboring peaks can be used both to normalize the moments measured at that peak and to add new (semi-local) features.

For the moment measurements to be stable with respect to image transformations, it is important to normalize them: to factor out the effect of the transformations. For example, orienting the moment measurements in the direction of the local gradient (the first moments) tends to cancel-out the effect of rotations in the image. However, since the local gradient can often have a low magnitude response, is not always possible to reliably use it to normalize all the moments at a given point. In fact, all of the derivatives (moments) have this problem. However, the angular position of a neighboring LoG peak relative to a given peak can often be stable and is used here to normalize the other features with respect to rotation. The relative distance of the neighbor to the given peak is also a measurement that can be exploited. In the current design, the angular position is used to normalize for rotation, and the distance as an additional feature to match. By using only immediate neighbors of a given peak and employing appropriate data structures for search efficiency, this semi-local analysis can be made fast and enhances the usefulness of moments.

In the current prototype, the first five moments (a row in Figure 10) where measured at every LoG peak, and the closest ten or so neighbors were used for semi-local features. For each pairing of a peak with one its neighbors, the moments of both where normalized by the angular position of the neighbor, and their distances and brightness differences were used as additional features. This gives us twelve semi-local features for every peak-neighbor pair. Since the measurements are occurring at peaks of the LoG, the LoG is naturally strong at these points. Thus the LoG magnitude is used to normalize the moments and the brightness difference with respect to changes in contrast.

Finally, since moments are measured only at LoG peaks, the second moments are correlated. This requires us to use their sum (LoG) and difference, rather than the second moments directly. Since the magnitude of everything is normalized using the LoG, this feature only contains sign information. However, since LoG peaks of both signs are being

selected and the magnitude tends to be strong at peaks, the sign information is very stable and useful.

The above allows us to describe every LoG peak in terms of a set of twelve-element feature vectors, one vector from each of the ten or so neighbors. Collectively, this set of features provides a fairly unique, stable and easy-to-compute representation of the image region about the peak. The reliability of the identification is furthered when these measurements are made at many peaks distributed about the image of the object.

5.2.2 Approximate match analysis

Given the above features computed over the image, or in regions of interest, the next step is to find likely matches between this data and similar, pre-stored features of the sought-for object. The present matching algorithm does not specifically require peaks of the LoG, thus the peaks of the previous subsection will be more generically referred to as points.

There may be many points (peaks) in both the model and current region of interest; therefore, searching through the set of possible permutations of point correspondences for the best match may be prohibitive. What is required is a match analysis that is fast, but sufficient to detect a strong match of the object when it is present, and little when it is not. Thus, an efficient method of computing an approximation of the best match is desired. In addition, the method must accommodate distortion: it cannot just rigidly translate the model, searching for the best model-to-image point alignment. The latter requirement is referred to here as flexible matching.

The simultaneous requirements of speed, sufficient evidence of match and flexible matching imply a matching process that starts with a very rough many-to-many point match and incrementally refines this match until the overall quality of the match can be used to determine if identification has been achieved. It is possible to efficiently design the computation of both the rough match and the incremental refinement, and the incremental refinement need only cycle until the overall confidence in the match is either sufficiently high (accept) or low (reject).

The rough matching step can be effectively implemented using efficient indexing and voting methods, and the refinement step can be effectively implemented using local match consistency filtering, followed by outlier analysis during the localization step. The last operation will be discussed in the next section.

More specifically, in the design presented and demonstrated here, the rough matching step proceeds as follows. Each detected point is associated with approximately ten neighboring detected points, and the twelve semi-local features defined in the last section are computed for each point-neighbor pair. For every detected point in the image region of interest, and for every one of its neighbors, use the twelve features to retrieve matching point-neighbor pairs of the model image that are stored in a data-base indexed by the same twelve features. For every model point-neighbor pair retrieved by the current image point-neighbor pair, vote for the match between the associated current image point and model image point. Point matches between the current image and the model image that receive multiple votes, in other words, have multiple neighbors with similar relative properties, are much more likely to be correct than those that do not. Thus, the overall,

initial rough match between the images is defined to be the set of point-to-point matches with vote counts above a threshold.

In the current implementation, the data-base of point-neighbor pairs is indexed by a binary number, where each bit of the number represents a binary quantization of a separate point-neighbor feature. For every model point-neighbor pair retrieved, its unquantized feature vector is correlated to the unquantized feature vector of the current image pair used to retrieve it. The correlations are normalized by the magnitude of the features, and correlations above 0.78 are accepted as voting point-neighbor pair matches. If any match between a model point and a current image point receives 4 or more votes, that match is added to the overall rough image match.

For the image pairs tested, there were typically 300 points per current image and 60 points per object model region. This gives a typical total of 18,000 possible point matches. The rough initial matching process typically produced a total of 100 point matches, or a reduction factor of 180. Since efficient implementations are possible for the feature detection, indexing and voting steps, the overall rough matching process should also be fast.

Even though this is a large reduction in point-to-point matches from the total possible, most of the matches are still erroneous. Since the number of point matches are used as a means of classifying the image, the initial point match set must be refined till most of the matches are likely to be correct. It is important to note that this does not mean that the resulting set of point matches has to be perfectly correct, just representative enough of the targeted match. For this reason, a refinement process is employed that is simple and fast, but may stop short of guaranteeing a completely correct point match.

The match selection criterion for each iteration of the refinement is analogous to the initial rough matching. The only difference is that the neighbor matches used in the voting step are required to have been previously accepted in the last iteration. Since they were accepted because they in turn are consistent with some minimal set of other neighbor matches, this iterative refinement tends to enforce approximate consistency over large sets of point matches and remove point matches not consistent with these large sets. Since only a small percentage of the total possible point matches are considered during the refinement step and the selection criterion involves a small number of neighbors, each refinement cycle is very fast.

Typically, it was found that the refinement process does a very good job of throwing out erroneous point matches. The total number of point matches typically starts at 100 and, after 2 or 3 iterations, it is reduced to approximately 15 to 20, with almost all being correct matches. This is true even for images of objects undergoing 3D rotations of up 45 degrees. For the design and demonstration discussed here, refinement is performed for 5 iterations.

5.2.3 Localization

Once a reasonable correspondence between model points and image points is produced, it can be evaluated by analyzing the fit of a parametrized image transformation to the point mapping. Given a sufficiently sophisticated parametrization and a method of detecting and removing inconsistent point matches (outliers), the number of remaining

point matches can be used as a criterion for accepting the overall match. Since the match procedure discussed above is restricted to semi-local match consistency, a global transformation fit analysis can often further reduce the chance of false identification. However, without the initial match analysis, the point match set would contain too many spurious matches to make localization with outlier analysis feasible.

For human heads undergoing changes in position and orientation relative to the camera, a six-parameter affine transformation is a reasonable, rough approximation of the point mapping from a model image to the given current image. This has been shown in frame-to-frame tracking using a fixed reference image and a range of head motions [7]. Given the experimental results presented below, it also seems sufficient for matching methods developed for re-acquisition and identification tasks.

The initial point matching step seems to output a low enough percentage of spurious point matches to support a relatively fast and straight-forward outlier analysis during transformation fitting. This will be especially true when the recognition module is operating in conjunction with the detection and tracking modules. These latter modules provide a rough segmentation of the image, removing distant parts of the image from consideration in the global match analysis. Given this combined initial filtering of point matches, it is feasible to employ M-Estimation as a method of determining point match consistency with a global transformation. Thus, a version of M-Estimation for affine transformation fitting was implemented that is analogous to that used in [9] for 3D object pose analysis.

The number of point matches remaining after the analysis was used to decide if the object is present in the image. The threshold for this number was determined empirically and depended on the total number of correct matches possible for a given object model image. The specific thresholds used are discussed in the feasibility analysis below.

5.3 Scientific contributions in recognition

In [14] and [11], methods of matching based on Gaussian-weighted moments are also presented. However, in both of these schemes and others utilizing moments, the moments of the object model image are measured only at a single or a few positions. The typical system depends on the measurement of a large, complex series of moments (up to 45) at these few, select positions for a sufficient set of discrimination features. This makes the match response very sensitive to occlusions, image clutter and other disruptions in the data at those few points. In [14], the large number of required Gaussian moments (45) was computed using nine derivatives measured over five octaves of scale. If the encoding is centered and scaled so that the image support for the largest scale is largely within the boundary of the object (i.e., extraneous data has low impact), then the support region for the smallest scale occupies only 1/256th of the object's region, and three-fifths of the features have support regions that are only one-sixteenth of the object region. This should make the system very sensitive to occlusion or other changes in a small, central area of the image patch.

A spatially distributed application of moments was developed here, followed by a process of using the features at all the measurement points in a global match analysis. In addition, a method is contributed that uses semi-local measurements (relative positions of

points) to normalize moment measurements and augment them with additional features.

Indexing and voting methods for detecting matches have been described in [10], [20], [13], and others. These methods have been applied to detecting rigid, well-modeled 3D structures in images using well-localized physical structures such as straight edges and elliptical arcs. In the current work, indexing and voting is applied as a first step in a process capable of flexibly matching objects with poorly known 3D structure directly to simple functions of the image data. The indexing and voting is done with respect to a strictly local and geometrically loose reference frame about the detected points. The detection of a global match transformation is not done during indexing; it is done afterwards, using a large fit-error tolerance. Thus, an approach to matching flexible, poorly modeled objects is contributed that has the potential speed of traditional indexing methods.

5.4 Study of feasibility

The two main requirements for feasibility are that the recognition success rate is good enough in the application operating conditions, and a real-time implementation is feasible. Given the domains of security and teleconferencing, an appropriate test of feasibility is the matching and discrimination of human heads undergoing normal motions.

This section has three parts. The first is a demonstration of the matching process in this context; the second is a demonstration of the use of the match response to discriminate multiple, similar objects; and the final part is a general discussion of the feasibility of satisfying the recognition requirements.

5.4.1 Matching under appearance variation

In teleconferencing and security situations, the head of a tracked subject can be expected to rotate, change in scale and position, and undergo deformations associated with talking and facial gestures. In the recognition studies described here, a single frame was used as a model image of the head. In practice, the recognition system will generally have more stored model images of the sought-for subject subject. However, there may often be ranges of views for which the subject is not modeled, thus it is important to demonstrate the ability to match over a range of views. In the important case of police photo files, there may only be full-face and profile views of a person; thus, a security system may have to recognize a face from a view that differs from the model views by as much as 45 degrees.

The matching output was considered useful if the great majority of matching peaks in the current image were in the head region and the estimated transformation given the matched peaks seemed reasonable. A quantitative study of discrimination rates based on this matching output is discussed in the next section.

The video data included rotations of the head of over forty-five degrees from the model view, and changes in object scale of over 24%. No limits were placed on where in the image the head could be, or how it was oriented in the image (rotation in the image plane). In fact, the performance of the system would have been the same if the head had been upside down. Three video sequences were studied, each containing a different

Figure 11: Examples of point matches. First column: model image. Other columns: matches between respective model image and other views of the object. Dots are LoG peaks that have been matched to model image peaks.

Figure 12: Examples of match-estimated object location. First column: model image showing nose location. Other columns: crosshair shows match-estimated location of model nose in other views of the object.

human subject undergoing motions. Two of the sequences covered over 20 seconds of continuous motion, and a total of 350 frames were used. In each sequence, a frontal view was selected as the model image and was matched to all frames in the sequence. The recognition process was applied to image data using a Gaussian scale (sigma) of 3.75, and all of the other system parameters were set to the values described above.

For almost all of the frames, the matched points were overwhelmingly concentrated in the head region and the estimated match transformation seemed reasonable. The latter was observed by using a crosshair to indicate the nose position predicted by the transformation. In almost every case, the crosshair was on or near the nose. Figure 11 shows examples of the matches. The first column shows the model image, and the other columns show matches between the model image and typical views of the object. The dots indicate LoG peaks in the current image that have been matched to the model image. In Figure 12, the crosshairs indicate the match-estimated location of the model image nose in the current image. The matching and localization of the objects seem reasonable in spite of the large changes in view and variations in facial features.

This demonstrates that the matching system is capable of tolerating view angle changes of a practical range. In addition, it handles changes in scale of up to 24%. As discussed in previous sections, the detection and tracking modules are capable of isolating regions of interest in every frame. The extent of these regions can be used as rough approximations of the scale of the object to be identified. Given these rough approximations of scale, the scale variation tolerance of 24% exhibited by the system demonstrates that practical performance levels are feasible here also.

5.4.2 Discrimination of similar objects

This section describes the quantitative study of the recognition system applied to discrimination of the similar objects. The objects were the three human heads shown in the previous section. Human heads tend to be structurally very similar. In fact, if rigid templates were used to match the heads, there may often be better alignment (lower match error) between the same views (e.g. frontal) of the different heads than between different views (e.g. frontal and three-quarter face) of the same head. This emphasizes the importance of using flexible, non-rigid models and matching methods for the purpose of identification, as was accomplished here.

In the discrimination experiment, the model image of every object was matched to every frame of every sequence described in the previous section. For each pair of model and current input image, the number of accepted peak matches was counted, and if the count was above some threshold, the modeled subject was considered identified in that input image. Since each model image had a different number of LoG peaks, a threshold was selected for each. Table 1 shows the thresholds and the resulting discrimination rates. The first column shows the matched peak count thresholds for each subject. The second column shows the percent of correct positives: the number of frames that were correctly identified as a given subject over the total number of frames of that subject. The third column shows the percent of false positives: the number of frames that were incorrectly identified as a given subject over the total number frames not containing that subject. The bottom row shows the averages for these statistics over all subjects studied;

Test subject	Required points	Correct positives	False positives
Subj A:	6	100%	0%
Subj B:	7	96%	3%
Subj C:	11	80%	4%
Average:	8	92%	2.3%

Table 1: Results of discrimination experiments after testing a total of 1,050 image pairs.

a total of 1,050 image pairs were tested.

As the table shows, the system is capable of correctly identifying an object 92% of the time and incorrectly identifies that object an average of only 2.3% of the time. This is between very similar objects and using only a single model frame and a single current input frame. In addition, the Gaussian moment computation is performed only at a single scale, while the images are of varying scale.

5.4.3 Feasibility of Recognition

A recognition system based on Gaussian-weighted moments, flexible and approximate matching, and robust object localization has been motivated, described and demonstrated. Given its performance on realistic data when only one model image and one input image is used, it seems reasonable to conclude that a usable level of performance for security and teleconferencing applications can be achieved when integrated with the detection and tracking modules. When embedded in an integrated system, the subject identification process would be able to utilize multiple model images of the same subject, gathered as it is being tracked. Tracking of the object also facilitates matching of the model data to multiple frames of the current video sequence. This multiplicity of data would greatly enhance the recognition rates that are already very good. In addition, other modalities such as color can be used to further identify subjects. Color can be particularly useful for rapid re-acquisition of a subject that has been recently tracked: a person's clothing should be unchanged in that time. Such situations frequently occur in teleconferencing and surveillance applications.

The recognition process can be efficiently implemented since it is constructed of a few components that are, in principal, very fast. The Gaussian and Laplacian filters have already been implemented to run in a fraction of a video frame period for the working version of the tracking system. The peaks of the Laplacian are simple to detect, and from that step on, the processing is restricted to pixels that are LoG peaks. This focuses the processing from tens of thousands of pixels to a few hundred. By utilizing small peak neighborhoods only, and by employing indexing and voting, the rough matching step requires a minimum of computation time. This is followed by a local match analysis that runs only 5 cycles over a much reduced set of potential point matches. Given this, and given the fact that the tracked regions isolated for recognition analysis are only a fraction of the image, it should be possible to perform all this computation in a fraction of a second, or fast enough to meet the requirement of 10 or so matches within two seconds.

6 Summary and conclusions

This report covers research in three areas of vision technology: object detection, tracking and recognition. Visual detection is the process of noting and locating something that is different or distinct from its surroundings in some way. For example, objects of interest are often moving relative to the background. Visual tracking is the process of re-locating a relevant object from frame to frame as it, or the camera, moves about. Visual recognition is the process of identifying some unknown portion of the image with something that has been visually acquired and modeled in the past.

These three functions are rapidly becoming feasible engineering goals. When integrated into a single system, their feasibility is enhanced and can be exploited in a variety of practical products that are otherwise difficult to realize.

6.1 Objectives and requirements

The focus of the current research effort is the development of vision technology for security and surveillance systems and teleconferencing. To this end, a system capable of detecting, tracking and identifying human subjects was developed.

Given the targeted applications, the design requirements for an automated assistant in this application areas are analyzed. The following system properties must be and have been demonstrated during the Phase I research:

- The detection process is sensitive enough to human motion to reliably detect people in conditions where the camera itself may be moving, as well as other objects.
- The position and extent estimates of the detected object are accurate enough to correctly focus recognition processing and frame the shots by controlling camera pan and tilt.
- Selected objects can be tracked when visible.
- The recognition system can identify an object under change in view. The recognition rate should be good enough under conditions of changing views that when the match results are intergrated across several frames, the recognition rate is very high.
- The processing for each module can be made real-time on conventional machines.

6.2 Design and scientific contributions

The design of the detection process is based on estimating the dominant visual motion in the image and extracting regions in the image that are not consistent with this motion. The analysis of the motion uses the correlation of the sign of the Laplacian of Gaussian (sLoG) filtered images. The contribution of this work is a design that explicitly compensates for the dominant motion and uses sLoG data, which is insensitive to contrast variation. A practical, real-time implementation using conventional hardware was also achieved.

The real-time tracking process successfully employs a unique combination of continued motion-based detection, an analysis of overlap of motion regions across different frames, and correlation-based tracking.

The extension of the detection and tracking methods by using local motion boundary detection is also explored. Evidence for motion boundaries can be found in the discontinuities of a densely sampled image motion field, in the appearance and disappearance of image texture along boundaries that undergo disocclusion or occlusion, and in the deformation of images along occluding boundaries. All of these types of evidence were exploited in the novel methods developed here.

A recognition module was designed that is an original combination of moment-based image representation, relational match analysis, efficient indexing and robust transformation estimation. A spatially distributed application of moments was developed, followed by a process of using the features at all the measurement points in a global match analysis. In addition, a method is contributed that uses semi-local measurements (relative positions of points) to normalize moment measurements and augment them by adding features.

Indexing and voting methods for recognition have been described elsewhere. However, these methods have been applied to detecting rigid, well-modeled 3D structures in images using well-localized physical structures such as straight edges and elliptical arcs. In the recognition design presented here, indexing and voting is applied as a first step in a process capable of flexibly matching objects with poorly known 3D structure directly to simple functions of the image data. The indexing and voting is done with respect to a strictly local and geometrically loose reference frame about the detected points. Thus, an approach to matching flexible, poorly modeled objects is contributed that has the potential speed of traditional indexing methods.

6.3 Feasibility of objectives

The detection and tracking module has been demonstrated to be very sensitive to human motion and capable of detecting human subjects even when the camera itself is moving. Detection and tracking quality were evaluated by using the computed position and extent information to control actual camera motion. The detection and tracking were considered competent if the camera motion could keep the subject of interest near the center of the image. The current version has been used to competently track people in real-time for periods of up to a half an hour (54,000 frames), and in the presence of multiple moving objects for up to twenty minutes (36,000 frames). The motions of the tracked subject can be quite large, covering the natural range, and also quite complex, including rapid 3D rotations. Also, the system has been tested for over a dozen subjects and many different backgrounds.

In addition, the extended study of local motion boundary detection has shown great promise, and should help in situations where dominant motion is difficult to measure, or the tracked figure is up against other, differently moving objects.

It has been demonstrated that the recognition module is capable of correctly identifying an object 92% of the time and incorrectly identifies an object only 2.3% of the time. This is against very similar objects and using only a single model frame and a

single current input frame. In the context of tracking, more model images and more current video images can be matched to greatly enhance the recognition rates that are already very good. In addition, other modalities such as color can be used to further identify subjects. Color can be particularly useful for rapid re-acquisition of a subject that had been recently tracked: a person's clothing should be unchanged in that time. Such situations frequently occur in teleconferencing and surveillance applications.

The detection and tracking modules have been demonstrated to run at real-time rates. The recognition process can be efficiently implemented since it is constructed of a few components that are, in principal, very fast. The Gaussian and Laplacian filters have already been implemented to run in a fraction of a video frame period for the working version of the tracking system. After detecting the LoG peaks, the processing is restricted to the few hundred LoG peaks, and the matching of these peaks is made efficient by employing indexing and voting. Once integrated with the motion-based detection and tracking system, the recognition module can be applied selectively to detected image regions, making it even more efficient.

6.4 Integration during Phase II

During the Phase II research, a prototype will be developed that integrates much of the designs developed in Phase I.

The local motion boundary analysis will be integrated with the detection and tracking module to enhance performance in situations discussed above.

Detection, tracking and recognition processes can interact in ways that enhance their performance, and this interaction will be exploited in the design of the final prototype in Phase II.

For example, recognition can clearly contribute to a more robust tracking by being used to re-acquire temporarily obscured objects. The current tracking design has been tested without it.

Recognition speed and performance can also benefit from integration with the other modules. By localizing the recognition processing to parts of the image associated with detected, but perhaps unknown, objects, the resources of the recognition system can be more efficiently allocated. Basically, the detected regions of interest could be used to register the pre-stored data of the lost objects, allowing the comparison to be done effectively. Even if the detected regions are not exact, the range of possible registrations could be limited by the detection process. In the current study, the recognition process was tested by simply applying it to the whole image and evaluating its ability to find the target object.

In addition, recognition can clearly benefit from tracking the object being matched. By tracking a detected region from frame-to-frame, the matching of this region to a sought-for object can be performed and integrated over several frames. As the object moves about, it will often be imaged from different views, and the subsequent series of matches will often converge to a confident identification.

Clearly, the integration of the different modules could greatly enhance their individual performances. This will be exploited during design of the complete prototype in Phase II.

References

[1] J. R. Bergen, P. Burt, R. Hingorani, and S. Peleg. Computing two motions from three frames. In *Proceedings of the Third International Conference on Computer Vision (ICCV90)*, pages 27–32, Osaka, Japan, 1990.

[2] M. Bichsel, "Segmenting simply connected moving objects in static scene", IEEE PAMI, vol 16(11), pp. 1138-1143, 1994.

[3] J. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(6):679–698, November 1986.

[4] J. Costeira and T. Kanade. A multi-body factorization method for motion analysis. In *Proceedings of the Fifth International Conference on Computer Vision (ICCV95)*, pages 1071–1076, Cambridge, MA, June 1995.

[5] D. A. Danielson. *Vectors and Tensors in Engineering and Physics.* Addison-Wesley, Reading, MA, 1992.

[6] A. Giachetti and V. Torre. Refinement of optical flow estimation and detection of motion edges. In B. Buxton and R. Cipolla, editors, *Computer Vision – ECCV96*, pages 151–160, Berlin, May 1996. Springer-Verlag.

[7] G. Hager and P. Belhumeur, "Real-time tracking of image regions with changes in geometry and illumination", *CVPR96*, pp. 403-410, 1996.

[8] M. Irani and P. Anandan. A unified approach to moving object detection in 2d and 3d scenes. In *Proceedings of the ARPA Image Understandig Workshop*, pages 707–718, Palm Springs, CA, February 1996.

[9] R. Kumar and A. Hanson, "Robust methods for estimating pose and a sensitivity analysis", *CVGIP: Image Understanding*, vol. 60, pp. 313-342, 1994.

[10] Y. Lamdan and H. J. Wolfson, "Geometric hashing: a general and efficient model-based recognition scheme", ICCV, pp. 238-249, 1988.

[11] T, Leung, M. Burl and P. Perona, "Finding faces in cluttered scenes using random labeled graph matching", ICCV95, p. 637, 1995.

[12] B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Proceedings of the 7th International Joint Conference on Artificial Intelligence (IJCAI81)*, 1981.

[13] C. Olson, "On the speed and accuracy of object recognition when using imperfect grouping", *Int. Symp on Comp. Vis.*, pp. 449 - 454, 1995.

[14] R. Rao and D. Ballard, "Object indexing using iconic sparse distributed memory", ICCV95, p.24, 1995.

[15] T. Reiss, *Recognizing planar objects using invariant image features*, Springer-Velag, 1993.

[16] H. Sawhney, S. Ayer, and M. Gorkani. Model-based 2d and 3d dominant motion estimation for mosaicing and video representation. In *Proceedings of IEEE International Conference on Computer Vision (ICCV95)*, pages 583–590, June 1995.

[17] J. Shi and C. Tomasi. Good features to track. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR94)*, pages 593–600, June 1994.

[18] M. Shizawa and K. Mase. A unified computational theory for motion transparency and motion boundaries based on eigenenergy analysis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR91)*, pages 289–295, June 1991.

[19] A. Spoerri and S. Ullman. The early detection of motion boundaries. In *Proceedings of the International Conference on Computer Vision (ICCV87)*, pages 209–218, 1987.

[20] D. Thompson and J. Mundy, "Three-dimensional model matching from an unconstrained viewpoint", *Int. Conf. on Rob. and Auto.*, pp. 208-220, 1987.

[21] M. A. Turk and A. P. Pentland, "Face recognition using eigenfaces", *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR91)*, pages 586–591, June 1991.

[22] J. Y. A. Wang and E. H. Adelson. Layered representation for motion analysis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR93)*, pages 361–366, New York, NY, June 1993.

[23] S. J. Wang and T. O. Binford. Generic, model-based estimation and detection of discontinuities in image surfaces. In *Proceedings of the ARPA Image Understanding Workshop*, pages 1443–1449, November 1994.